

Van Leeuwen & Van Leeuwen
Attorneys at Law
6123 Pebble Garden Court
Austin, Texas 78739
Phone # 512-301-6738
FAX # 512-301-6742

RECEIVED
CENTRAL FAX CENTER

NOV 21 2005

DATE: November 21, 2005

Number of Pages to Follow (including this cover sheet) 37

SEND TO: United States Patent Office

Examiner: **Vu, Tuan A.**

Group Art Unit: **2193**

Tel No: 571-272-3735

Fax #: 571-273-8300

FROM: Leslie A. Van Leeuwen
Van Leeuwen & Van Leeuwen
Registered Patent Attorneys
6123 Pebble Garden Court
Austin, Texas 78739
Tel No: 512-301-6738
Fax No. 512-301-6742

THIS MESSAGE IS INTENDED ONLY FOR THE USE OF THE INDIVIDUAL OR ENTITY TO WHICH IT IS ADDRESSED, AND MAY CONTAIN INFORMATION THAT IS PRIVILEGED, CONFIDENTIAL AND EXEMPT FROM DISCLOSURE UNDER APPLICABLE LAW. IF THE READER OF THIS MESSAGE IS NOT THE INTENDED RECIPIENT, OR THE EMPLOYEE OR AGENT RESPONSIBLE FOR DELIVERING THE MESSAGE TO THE INTENDED RECIPIENT, YOU ARE HEREBY NOTIFIED THAT ANY DISSEMINATION, DISTRIBUTION OR COPYING OF THIS COMMUNICATION IS STRICTLY PROHIBITED. IF YOU HAVE RECEIVED THIS COMMUNICATION IN ERROR, PLEASE NOTIFY US IMMEDIATELY BY TELEPHONE AND RETURN THE ORIGINAL MESSAGE TO US AT THE ADDRESS ABOVE VIA THE U.S. POSTAL SERVICE. THANK YOU.

Docket No. RSW920010048US1

Serial No. **10/047,792**

Atty: GRW / JVL

Applicant: Barker, et al.

<input checked="" type="checkbox"/> Transmittal Letter (2 copies)	<input checked="" type="checkbox"/> Certificate of Facsimile (incl. w/Brief)
<input type="checkbox"/> Amendment (<input type="text"/> pages)	<input type="checkbox"/> Notice of Appeal
<input type="checkbox"/> Amendment AF	<input checked="" type="checkbox"/> Appeal Brief (34 pages)
<input type="checkbox"/> Ext. of Time	<input type="checkbox"/> Reply Brief
<input type="checkbox"/> IDS Statement	<input type="checkbox"/> Change of Address
<input type="checkbox"/> Other	

Deposit Acct. No. 09-0461

Fees: Amendment Notice of Appeal Appeal Brief \$500.00 Other

File: **R108**

COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

IBM DOCKET NO. RSW920010048US1
DATE: November 21, 2005

Application Serial No.: 10/047,792

Sir:

Assignee Name: International Business Machines Corporation
Assignee Residence: Armonk, New York

Transmitted herewith for filing is the Patent Application of:

Inventors: Barker, et al.

For: System and Method for Converting Management Models to Specific Console Interfaces

Enclosed are:

X Appeal Brief

Any additional filing fees have been calculated as shown below:

For	Number Filed	Number Added	Number Extra	Rate	Additional Fee
Total Claims	0	0	0	x \$50	\$ 0.00
Indep. Claims	0	0	0	x \$200	\$ 0.00

Total: \$ 0.00

X Please charge my Deposit Account No. 09-0461 in the amount of \$500.00 per 37 CFR § 41.20(b)(2). A duplicate copy of this sheet is enclosed.

X The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 09-0461. A duplicate copy of this sheet is enclosed.

X Any additional filing fees required under 37 CFR Sect. 1.16.

X Any patent application processing fees under 37 CFR Sect. 1.17.

X No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and the undersigned hereby authorizes the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0461.

Respectfully submitted,

By Leslie A. Van Leeuwen
Leslie A. Van Leeuwen, Reg. No. 42,196
Van Leeuwen & Van Leeuwen
Attorneys for Applicant
Telephone: (512) 301-6738
Facsimile: (512) 301-6742

Atty Ref. No. R108

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED
CENTRAL FAX CENTER

NOV 21 2005

In re application of:
Barker, et. al.

\$ Group Art Unit: 2193

\$

\$ Examiner: Vu, Tuan A.

\$

Serial No.: 10/047,792

\$ Attorney Docket No.

Filed: January 14, 2002

\$ RSW920010048US1

\$

Title: System and Method for
Converting Management
Models to Specific Console
Interfaces

\$ IBM Corporation

\$ Software Group IP Law --

\$ T81/503

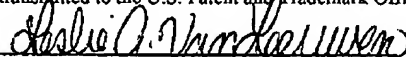
\$ 3039 Cornwallis Road

\$ Research Triangle Park, NC

\$ 27709

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450**Certificate of Mailing or Transmission**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 or facsimile transmitted to the U.S. Patent and Trademark Office on the date shown below.


Leslie A. Van Leeuwen11/21/2005
Date**APPELLANTS' BRIEF (37 CFR § 41.37)**

Sir:

A. INTRODUCTORY COMMENTS

This brief is filed in support of the previously filed Notice of Appeal, filed in this case on September 27, 2005, which appealed from the decision of the Examiner dated June 27, 2005, finally rejecting claims 1-32. Please charge the required fee under 37 CFR § 41.20(b)(2) to IBM Corporation Deposit Account No. 09-0461.

The two-month deadline for filing this Appeal Brief is November 28, 2005 (note that November 27 is a Sunday), therefore, no extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and the undersigned hereby authorizes the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0461.

Docket No. RSW920010048US1

Page 1 of 34

Atty Ref. No. R108

11/22/2005 JBALINAN 00000014 090461 10047792

Barker, et. al. - 10/047,792

01 FC:1402 500.00 DA

PATENT

B. REAL PARTY IN INTEREST

The real party in interest in this appeal is International Business Machines Corporation, which is the assignee of the entire right, title, and interest in the above-identified patent application.

C. RELATED APPEALS AND INTERFERENCES

With respect to other prior or pending appeals, interferences, or judicial proceedings that are related to, will directly affect, be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such prior or pending appeals, interferences, or judicial proceeding known to Appellants, Appellants' legal representative, or assignee.

D. STATUS OF CLAIMS*1. Total number of claims in application*

There are 32 claims pending. Eight claims are independent claims (1, 10, 19, 28, 29, 30, 31, and 32), and the remaining claims are dependent claims.

2. Status of all claims in application

- Claims canceled: none
- Claims withdrawn from consideration but not canceled: none
- Claims pending: 1-32
- Claims allowed: None
- Claims rejected: 1-32

3. Claims on appeal

The claims on appeal are: 1-32.

PATENT

E. STATUS OF AMENDMENTS

All amendments have been entered in this case. No amendments have been made to the claims after the Final Office Action.

F. SUMMARY OF CLAIMED SUBJECT MATTER

Appellants provide a concise summary of the claimed subject matter as follows. Claims 1, 10, 19, 28, 29, 30, 31, and 32 are independent claims. Note that claims 1-9, 28, and 29 are method claims, claims 10-18, and 30 are information handling system claims, and claims 19-27, 31, and 32 are computer program product claims. Independent claims 10, 19, 30, 31, and 32 include logic elements and means plus function limitations that correspond to the method steps set forth in independent claims 1, 28, and 29. An information handling system capable of implementing Appellants' invention, as claimed in independent claims 10 and 30, is shown in Figure 18, and described in Appellants' specification on page 45, line 10 through page 46, line 24. Support for independent computer program product claims 19, 31, and 32 is described in Appellants' specification on page 46, line 25 through page 47, line 14. In addition, support for each of the method steps, logic elements, and means plus function limitations of the independent claims are discussed below. The specific citations to Appellants' Figures and Specification are meant to be exemplary in nature, and do not limit the scope of the claims. In particular, the citations below do not limit the scope of equivalents as provided under 35 U.S.C. § 112, sixth paragraph.

As claimed in independent claims 1, 10, and 19, Appellants claim a method, information handling system, and computer program product for converting management models to one or more console interfaces, including receiving a console selection corresponding to one of the console interfaces (see e.g. Figure 3, reference numeral 345, page 15, line 6 through page 16, line 13), identifying one or more console algorithms corresponding to the console selection (see e.g. Figure 3, reference numeral 350, page 15, line 6 through page 16, line 13), retrieving a generic management object from a management definition object (see e.g. Figure 5, reference numeral 500, page 20, line 9 through page 21, line 17), and creating an extension file adapted to perform the generic management object on the selected console interface (see e.g. Figure 3, reference numerals 360 and 365, page 15, line 6 through page 16, line 13).

PATENT

As claimed in independent claim 28, Appellants claim a method for converting management models to one or more console interfaces, including the following elements:

receiving a console selection corresponding to one of the console interfaces (see, e.g.

Figure 3, reference numeral 345, page 15, line 6 through page 16, line 13);

identifying one or more console algorithms corresponding to the console selection (see

e.g. Figure 3, reference numeral 350, page 15, line 6 through page 16, line 13);

retrieving a generic management object from a management definition object (see e.g.

Figure 5, reference numeral 500, page 20, line 9 through page 21, line 17);

creating an extension file adapted to perform the generic management object on the

selected console interface (see e.g. Figure 3, reference numerals 360 and 365,

page 15, line 6 through page 16, line 13);

identifying one or more object classes and one or more object associations that include

the object classes (see e.g. Figure 11, reference numeral 1110, page 32, line 5

through page 33, line 22);

selecting one of the object classes (see e.g. Figure 11, reference numerals 1105 and 1125,

page 32, line 5 through page 33, line 22);

selecting one of the associations that include the object class, wherein the association

includes a simple association (see e.g. Figure 11, reference numeral 1150, page

32, line 5 through page 33, line 22);

identifying one or more target classes corresponding to the selected association (see e.g.

Figure 11, reference numerals 1130 and 1190, page 32, line 5 through page 33,

line 22) ;

locating one or more second associations corresponding to the identified target class (see

e.g. Figure 11, reference numerals 1160 and 1170 page 32, line 5 through page

33, line 22); and

determining a layout format based on the number of located second associations (see e.g.

Figure 10, reference numerals 1040 - 1055, page 29, line 6 through page 32, line

4) .

PATENT

As claimed by Appellants in independent claim 29, Appellants claim a method for converting management models to one or more console interfaces, including the following elements:

receiving a console selection corresponding to one of the console interfaces (see, e.g. Figure 3, reference numeral 345, page 15, line 6 through page 16, line 13);
identifying one or more console algorithms corresponding to the console selection (see e.g. Figure 3, reference numeral 350, page 15, line 6 through page 16, line 13);
retrieving a generic management object from a management definition object (see e.g. Figure 5, reference numeral 500, page 20, line 9 through page 21, line 17);
creating an extension file adapted to perform the generic management object on the selected console interface (see e.g. Figure 3, reference numerals 360 and 365, page 15, line 6 through page 16, line 13);
retrieving user interface data from the management definition object (see e.g. Figure 9, reference numerals 940 - 970, page 27, line 5 through page 29, line 5) ;
converting the user interface data to one or more national languages (see e.g. Figure 14, reference numeral 1430, page 36, line 22 through page 38, line 14);
storing the converted user interface data in one or more national language files (see e.g. Figure 14, reference numerals 1435 - 1445, page 36, line 22 through page 38, line 14);
selecting one of the national languages (see e.g. Figure 15, reference numerals 1504, 1505 and 1590, page 38, line 15 through page 40, line 20) ;
retrieving one of the national language files corresponding to the selected national language (see e.g. Figure 15, reference numerals 1540 and 1560, page 38, line 15 through page 40, line 20); and
generating one or more graphical user interfaces using the converted user interface data retrieved from the national language file (see e.g. Figure 9, reference numeral 980, page 27, line 5 through page 29, line 5) .

As claimed by Appellants in independent claim 30, Appellants claim an information handling system, including the following elements:

PATENT

one or more processors (see e.g. Figure 18, reference numeral 1800, page 45, line 10 through page 46, line 24);

a memory accessible by the processors (see e.g. Figure 18, reference numeral 1820, page 45, line 10 through page 46, line 24);

a nonvolatile storage area accessible by the processors (see e.g. Figure 18, reference numerals 1850 and 1872, page 45, line 10 through page 46, line 24); and

a conversion tool for converting generic management data to specific console interfaces, the conversion tool including:

receiving logic for receiving a console selection corresponding to one of the console interfaces (see, e.g. Figure 3, reference numeral 345, page 15, line 6 through page 16, line 13);

identification logic for identifying one or more console algorithms corresponding to the console selection (see e.g. Figure 3, reference numeral 350, page 15, line 6 through page 16, line 13);

retrieval logic for receiving a generic management object from a management definition object (see e.g. Figure 5, reference numeral 500, page 20, line 9 through page 21, line 17); and

creation logic for creating an extension file adapted to perform the generic management object on the selected console interface (see e.g. Figure 3, reference numerals 360 and 365, page 15, line 6 through page 16, line 13);

identification logic for identifying one or more object classes and one or more object associations that include the object classes (see e.g. Figure 11, reference numeral 1110, page 32, line 5 through page 33, line 22);

selection logic for selecting one of the object classes and for selecting one of the associations that include the object class, wherein the association includes a simple association (see e.g. Figure 11, reference numerals 1105 and 1125, page 32, line 5 through page 33, line 22);

identification logic for identifying one or more target classes corresponding to the selected association (see e.g. Figure 11, reference numerals 1130 and 1190, page 32, line 5 through page 33, line 22);

PATENT

location logic for locating one or more second associations corresponding to the identified target class (see e.g. Figure 11, reference numerals 1160 and 1170 page 32, line 5 through page 33, line 22);

determination logic for determining a layout format based on the number of located second associations (see e.g. Figure 10, reference numerals 1040 - 1055, page 29, line 6 through page 32, line 4); and

display logic for displaying the target class name in response to determining that there is one second association and for displaying the association role in response to determining that there are more than one second associations (see e.g. Figure 10, reference numerals 1040 - 1055, page 29, line 6 through page 32, line 4).

As claimed by Appellants in independent claim 31, Appellants claim a computer program product, including the following elements:

- means for receiving a console selection corresponding to one of the console interfaces (see, e.g. Figure 3, reference numeral 345, page 15, line 6 through page 16, line 13);
- means for identifying one or more console algorithms corresponding to the console selection (see e.g. Figure 3, reference numeral 350, page 15, line 6 through page 16, line 13);
- means for retrieving a generic management object from a management definition object (see e.g. Figure 5, reference numeral 500, page 20, line 9 through page 21, line 17);
- means for creating an extension file adapted to perform the generic management object on the selected console interface (see e.g. Figure 3, reference numerals 360 and 365, page 15, line 6 through page 16, line 13);
- means for identifying one or more object classes and one or more object associations that include the object classes (see e.g. Figure 11, reference numeral 1110, page 3, line 5 through page 33, line 22);
- means for selecting one of the object classes (see e.g. Figure 11, reference numerals 1105 and 1125, page 32, line 5 through page 33, line 22);

PATENT

means for selecting one of the associations that include the object class, wherein the association includes a simple association (see e.g. Figure 11, reference numeral 1150, page 32, line 5 through page 33, line 22);

means for identifying one or more target classes corresponding to the selected association (see e.g. Figure 11, reference numerals 1130 and 1190, page 32, line 5 through page 33, line 22);

means for locating one or more second associations corresponding to the identified target class (see e.g. Figure 11, reference numerals 1160 and 1170 page 32, line 5 through page 33, line 22);

means for determining a layout format based on the number of located second associations (see e.g. Figure 10, reference numerals 1040 - 1055, page 29, line 6 through page 32, line 4);

means for displaying the target class name in response to determining that there is one second association (see e.g. Figure 10, reference numerals 1094, page 29, line 6 through page 32, line 4) and

means for displaying the association role in response to determining that there are more than one second associations (see e.g. Figure 10, reference numerals 1040 - 1055, page 29, line 6 through page 32, line 4).

As claimed by Appellants in independent claim 32, Appellants claims a computer program product, including the following elements:

means for receiving a console selection corresponding to one of the console interfaces (see, e.g. Figure 3, reference numeral 345, page 15, line 6 through page 16, line 13);

means for identifying one or more console algorithms corresponding to the console selection (see e.g. Figure 3, reference numeral 350, page 15, line 6 through page 16, line 13);

means for retrieving a generic management object from a management definition object (see e.g. Figure 5, reference numeral 500, page 20, line 9 through page 21, line 17);

PATENT

means for creating an extension file adapted to perform the generic management object on the selected console interface (see e.g. Figure 3, reference numerals 360 and 365, page 15, line 6 through page 16, line 13);

means for creating a tree node displayable on a management console corresponding to the selected console interface (see e.g. Figure 16, page 40, line 21 through page 43, line 17) ;

means for creating a popup menu corresponding to the tree node, wherein the popup menu includes one or more menu selections (see e.g. Figure 12, reference numeral 1205, page 33, line 23 through page 35, line 13) ; and

means for associating a method to each of the menu selections (see e.g. Figure 12, reference numeral 1220, page 33, line 23 through page 35, line 13) .

Support for each of Appellants' means plus function limitations set forth in dependent claims that are argued separately is provided below. Note that general support for an information handling system and computer program product is discussed above. The specific citations to Appellant's Figures and Specification are meant to be exemplary in nature, and do not limit the scope of the claims, as provided under 35 U.S.C. § 112, sixth paragraph.

Claim 25 is argued below as part of a separate group including claims 7, 16, and 29. Claim 25 includes the following means plus function limitations:

means for retrieving user interface data from the management definition object (see e.g. Figure 9, reference numerals 940 - 970, page 27, line 5 through page 29, line 5);

means for converting the user interface data to one or more national languages (see e.g. Figure 14, reference numeral 1430 page 36, line 22 through page 38, line 14);

means for storing the converted user interface data in one or more national language files (see e.g. Figure 14, reference numerals 1435 - 1445, page 36, line 22 through page 38, line 14);

means for selecting one of the national languages (see e.g. Figure 15, reference numerals 1504, 1505 and 1590, page 38, line 15, through page 40, line 20);

PATENT

means for retrieving one of the national language files corresponding to the selected national language (see e.g. Figure 15, reference numerals 1540 and 1560, page 38, line 15 through page 40, line 20); and

means for generating one or more graphical user interfaces using the converted user interface data retrieved from the national language file (see e.g. Figure 9, reference numeral 980, page 27, line 5 through page 29, line 5).

G. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 3, 9, 10, 12, 18, 19, 21, and 27-32 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Appellants regard as the invention. In particular, the Examiner has rejected these claims under 35 U.S.C. § 112, second paragraph, due to the limitation "adapted to" found in each of these claims.

Claims 1-6, 8-15, 17-24, 26-28, and 30-32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Patrizio, U.S. Publication No. 2003/0095145 (hereinafter Patrizio) in view of Agnihotri et al., U.S. Patent No. 6,311,321 (hereinafter Agnihotri). Claims 7, 16, 25, and 29 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Patrizio and Agnihotri in view of Cramon et al., U.S. Publication No. 2002/0103660 (hereinafter Cramon).

H. ARGUMENT**1. Claims 1, 3, 9, 10, 12, 18, 19, 21, And 27-32 Are Not Indefinite Under 35 U.S.C. § 112, Second Paragraph**

The Examiner has rejected claims 1, 3, 9, 10, 12, 18, 19, 21, and 27-32 due to the use of the term "adapted to" in these claims. In particular, the Examiner states that the term "adapted to" does not specify the metes and bounds of a feature of the invention, because the term "does not define whether an action is taken or not" (see Final Office Action, page 2, line 21 through page 3, line 2). Appellants respectfully disagree that use of the term "adapted to" renders the claim language indefinite. As stated in the Manual of Patent Examining Procedure, section 2173.05(g) (emphasis added):

PATENT

2173.05(g) Functional Limitations [R-3]

A functional limitation is an attempt to define something by what it does, rather than by what it is (e.g., as evidenced by its specific structure or specific ingredients). **There is nothing inherently wrong with defining some part of an invention in functional terms. Functional language does not, in and of itself, render a claim improper.** *In re Swinehart*, 439 F.2d 210, 169 USPQ 226 (CCPA 1971).

A functional limitation must be evaluated and considered, just like any other limitation of the claim, for what it fairly conveys to a person of ordinary skill in the pertinent art in the context in which it is used. **A functional limitation is often used in association with an element, ingredient, or step of a process to define a particular capability or purpose that is served by the recited element, ingredient or step.** >In *Innova/Pure Water Inc. v. Safari Water Filtration Sys. Inc.*, 381 F.3d 1111, 1117-20, 72 USPQ2d 1001, 1006-08 (Fed. Cir. 2004), the court noted that the claim term "operatively connected" is "a general descriptive claim term frequently used in patent drafting to reflect a functional relationship between claimed components," that is, the term "means the claimed components must be connected in a way to perform a designated function." "In the absence of modifiers, general descriptive terms are typically construed as having their full meaning." *Id.* at 1118, 72 USPQ2d at 1006. In the patent claim at issue, "subject to any clear and unmistakable disavowal of claim scope, the term 'operatively connected' takes the full breath of its ordinary meaning, i.e., 'said tube [is] operatively connected to said cap' when the tube and cap are arranged in a manner capable of performing the function of filtering." *Id.* at 1120, 72 USPQ2d at 1008.<

Appellants note that the term "adapted to" is used as a functional limitation in claims 1, 3, 9, 10, 12, 18, 19, 21, and 27-32. In other words, the term "adapted to" is used in association with an element of the claim to define a particular capability or purpose that is served by the recited element. For example, claim 1 includes the element of "creating an extension file adapted to perform the generic management object on the selected console interface." In claim 1, Appellants are claiming the *creating* of an extension file. The created extension file is adapted to perform the generic management object on the selected console interface at some point after its creation. Thus, the "adapted to" language explains the purpose for creating the extension file, in other words, it explains what the extension file will be used for after it is created. The extension file does not perform the generic management object on the selected console interface at the time it is created, and thus it would not make sense to include this specific action in claim 1. Claim 1 is concerned with creating the extension file, and, at the time of the extension file's creation, it is

PATENT

not used to perform the generic management object on the selected console interface. However, the extension file is "adapted to" perform the generic management object on the selected console interface at some future point in time.

For the reasons set forth above, Appellants respectfully submit that the use of the term "adapted to" does not render claims 1, 3, 9, 10, 12, 18, 19, 21, and 27-32 indefinite, and respectfully request that the rejection under 35 U.S.C. § 112, second paragraph, be withdrawn.

2. Claims 1-6, 8-15, 17-24, and 26-27 Are Not Obvious And Are Therefore Patentable Over Patrizio In View Of Agnihotri

Independent claims 1, 10, and 19 claim a method, system, and program product for converting management models to one or more console interfaces. Using claim 1 as an exemplary claim, each of these independent claims includes the limitations of:

- receiving a console selection corresponding to one of the console interfaces;
- identifying one or more console algorithms corresponding to the console selection;
- retrieving a generic management object from a management definition object; and
- creating an extension file adapted to perform the generic management object on the selected console interface.

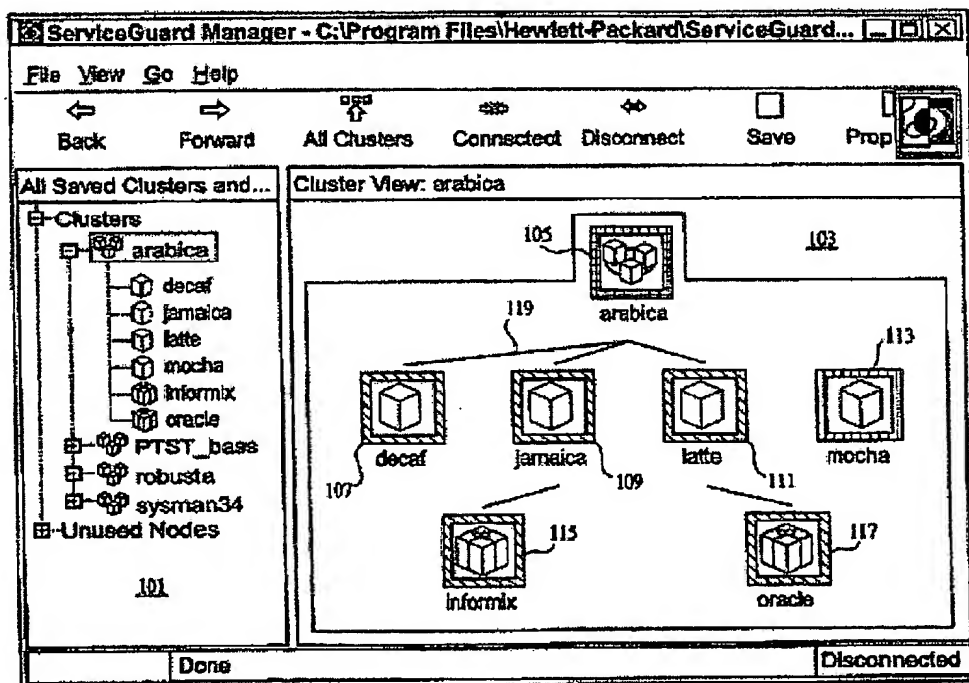
The Final Office Action mistakenly contends that Patrizio and Agnihotri teach or suggest each of these limitations. Appellants respectfully disagree.

First, regarding the limitation of *receiving a console selection corresponding to one of the console interfaces*, the Final Office Action does not even recite the limitation as claimed. Instead, the Final Office Action states that Patrizio teaches "receiving a selection corresponding to one of the graphical user interface panels" (see Final Office Action, page 3, lines 16-17). In essence, the Final Office Action is rejecting a claim limitation that has not been claimed by Appellants. A "console interface," as used in Appellants' disclosure is an interface provided by a "management console" which "allow common applications from which a user views and manipulates data associated with a software program or device that is accessible by the console." (see, e.g., Appellants' specification at page 3, line 27 to page 4, line 13). Appellants simply are not claiming "receiving a selection corresponding to one of the graphical user interface panels," as suggested by the Final Office Action's rejection. Appellants clearly teach and claim receiving

PATENT

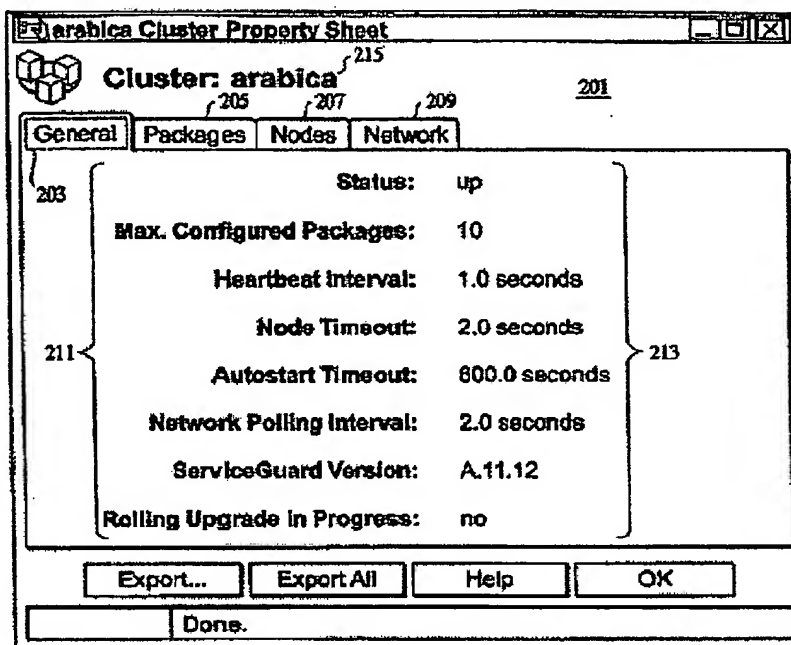
“a console selection,” that corresponds to “one of the console interfaces,” not a “selection” that corresponds to “one of the graphical user interface panels.” Appellants respectfully assert that Patrizio does not teach or suggest “receiving a console selection corresponding to one of the console interfaces,” as taught and claimed by Appellants.

The Final Office Action cites Figures 1 and 2 from Patrizio’s reference to support this unfounded rejection. These figures are reproduced below. Figure 1 shows a top-level screen shot from a “ServiceGuard Manager” product, and Figure 2 shows a screen shot of a “General” tab from a cluster property sheet. A review of these figures reveals that neither teaches or suggests a “console selection corresponding to one of the console interfaces.” In fact, while Appellants’ claims are directed to “console interfaces” (explained thoroughly in Appellants’ specification), the term “console” never even appears in the Patrizio reference.



Patrizio's Figure 1

PATENT



Patrizio's Figure 2

Moreover, the "ServiceGuard" product discussed in Patrizio is described as being a "specialized facility for protecting mission-critical applications from a wide variety of hardware and software failures," and is not described as being a management console. In addition, Appellants have reviewed the Patrizio reference and found that Patrizio does not teach or suggest "plug-ins" or ways to extend the functionality of the ServiceGuard product described by Patrizio. Extending the console using plug-in, or extension, files is a key aspect of Appellants' claimed invention and is completely missing from Patrizio.

The Final Office Action contends that Patrizio teaches Appellants' next limitation, "identifying one or more console algorithms corresponding to the console selection," citing various sections of Patrizio that teach structuring a managed object format (MOF) file using a unified modeling language (UML) class diagram. While Appellants' management definition object is similar to a CIM MOF file (see Appellants' claim 2), the point of the "identifying..." limitation is to identify console algorithms that correspond to the selected console selection. Because Patrizio does not teach or suggest "console selections," it logically follows that Patrizio cannot teach identifying algorithms that correspond to the console selection. In addition, the

PATENT

cited reference on page 3, paragraph 42 of Patrizio, actually teaches away from Appellants' claimed limitation. The last sentence of this paragraph states: "Inside of a file that is internal to ServiceGuard Manager there is a MOF file which contains a description telling the program how a cluster property sheet should be rendered." (emphasis added). While Appellants' claimed method is for creating extension files that can perform a management object that can be used with a console interface, Patrizio appears to be teaching a system where the rendering description is "internal to" a system and, consequently, teaches away from Appellants' claimed method, system, and program product that provides for the extension of the management object onto the selected console interface.

Next, the Final Office Action contends that Patrizio teaches or suggests Appellants' claimed limitation of "retrieving a generic management object from a management definition object," citing paragraph 41 of Patrizio. Here, Patrizio does teach using property sheets and Patrizio's property sheets are described in the MOF file. However, the shortcoming of Patrizio is not with its use of a management object from a management definition file, it is that Patrizio does not teach or suggest creating an "extension file adapted to perform the generic management object on [a] selected console interface," as taught and claimed by Appellants.

The Final Office Action contends that Patrizio teaches Appellants' limitation of "creating an extension file adapted to perform the generic management object on the selected console interface," citing Figure 9A of Patrizio. According to Patrizio, Figure 9A shows "a class schema defining a layout for property sheets." The Final Office Action notes "(Fig. 9A – Note: enlisting area wherein deriving additional sheets -- with JAVA class extended functionalities to support a GUI panel – takes place reads on extension file) (see Final Office Action, page 3, line 25 through page 4, line 2). However, even if Patrizio teaches a JAVA class that supports a GUI panel "on the selected graphical user interface," this is not the limitation claimed by Appellants. Appellants' extension file is adapted to work with a selected console interface. On the other hand, the program taught by Patrizio does not teach or suggest integration with external programs using plug-ins or extension files. Instead, as described above, Patrizio teaches a system where such information is kept "internal to" the program. Moreover, while the GUI described in the property sheets can be changed, Patrizio only teaches using the property sheets to turn on/off aspects (e.g., tabs) of the GUI when the ServiceManager program is executed (see

PATENT

Patrizio, paragraph 45, top of page 4). Again, this teaches away from “receiving a console selection...” as there is only one management program (“ServiceManager”) being taught by Patrizio. In addition, this teaches away from Appellants’ claimed element of a “generic management object.” Instead, Patrizio appears to teach more specific management objects that are designed and created for interaction with a specific software tool (i.e. Patrizio’s “ServiceManager” program).

The Final Office Action admits that Patrizio does not teach or suggest that “the selected graphical user interface panel is a console interface; nor does Patrizio explicitly teach a selection corresponding to one of said console interfaces” (see Final Office Action, page 4, lines 4-6). Despite this serious admitted shortcoming of Patrizio, the Final Office Action further states that “the limitation of selection corresponding to a console interface is disclosed via Patrizio’s selection of panel components to modify a certain GUI layout” (see Final Office Action, page 4, lines 6-9). Appellants respectfully disagree. As explained in detail above, the selection of some GUI properties in a property sheet is entirely different from selection of a console interface. The GUI selection only determines graphical qualities of the display when viewed on one particular system (e.g., the SystemManager program). On the other hand, Appellants’ selection of a particular console interface allows Appellants’ claimed extension files to be created in a manner so that they perform management functions from the selected console interface. Nowhere does Patrizio teach or suggest that the management functions, such as the GUI panels described in Patrizio’s property sheets, are created in a manner, based on a console selection, that allow the management functions to be performed on a different console. Instead, the management functions described by Patrizio are taught as being exclusively performed using Patrizio’s “ServiceManager” software. In addition, Appellants have pointed out various sections of Patrizio that describe how Patrizio’s property sheets are rendered using data that is “internal to” Patrizio’s ServiceManager software. Patrizio does not teach or suggest using such “internal data” for use by a different program (e.g., by a management console program such as the Microsoft Management Console (MCC), the AS/400 System Console, the Tivoli Console, or other commonly used management consoles).

To overcome the admitted shortcomings of Patrizio, the Final Office Action contends that Agnihotri discloses the selection of a console, citing col. 5, lines 1-18 of Agnihotri. The cited

PATENT

section of Agnihotri discloses a “wizard application” that is used to install a product onto a selected management console. However, Agnihotri does not teach or suggest Appellants’ claimed limitations of “identifying one or more console algorithms corresponding to the console selection,” “retrieving a generic management object from a management definition object,” or “creating an extension file adapted to perform the generic management object on the selected console interface.” Indeed, the Final Office Action does not contend that Agnihotri teaches such limitations. What Agnihotri is describing is a well known operation performed when installing a product with a given management console. However, Agnihotri’s install files and configuration files are directed to the management console selected by the user. Agnihotri does not teach or suggest using “algorithms” and “generic management objects” that are used to create “an extension file adapted to perform the generic management object on the selected console interface,” as claimed by Appellants.

Neither Patrizio nor Agnihotri, either alone or in combination, teach or suggest the elements of claims 1, 10, and 19. Therefore, Appellants respectfully submit that independent claims 1, 10, and 19, and the claims which depend from them, are patentable over Patrizio in view of Agnihotri.

In addition, Appellants respectfully remind the Examiner that MPEP § 2141 sets forth, among other things, basic consideration that apply to obviousness rejections under 35 U.S.C. § 103. MPEP § 2141 states as follows:

BASIC CONSIDERATIONS WHICH APPLY TO OBVIOUSNESS REJECTIONS

When applying 35 U.S.C. 103, the following tenets of patent law must be adhered to:

- (A) The claimed invention must be considered as a whole;
- (B) The references must be considered as a whole and must suggest the desirability and thus the obviousness of making the combination;
- (C) The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention; and
- (D) Reasonable expectation of success is the standard with which obviousness is determined.

Hodosh v. Block Drug Co., Inc., 786 F.2d 1136, 1143 n.5, 229 USPQ 182, 187 n.5 (Fed. Cir. 1986)(emphasis added).

PATENT

Appellants respectfully submit that Appellants' claims were not considered as a whole in the Final Office Action. As described above, the Final Office Action rejected some of Appellants' claim limitations by improperly substituting terms found in Patrizio with Appellants' claim limitations, even though the terms and the limitations are not similar or interchangeable. In particular, the Final Office Action describes Appellants' first limitation as "receiving a *selection* corresponding to one of *the graphical user interface panels*," while the actual claim limitation reads "receiving a *console selection* corresponding to one of *the console interfaces*." As described above, Appellants' "console selection" is not analogous to Patrizio's "selection" and Appellants' "console interfaces" is nothing like Patrizio's "graphical user interface panels." Appellants' console interfaces correspond to particular management consoles and are not merely graphical user interface panels.

Second, Appellants pointed out how Patrizio's teaching is for GUI elements that correspond to a particular software program. Patrizio does not teach or suggest a desirability to have the property sheets and GUI interfaces taught in Patrizio to work with other software programs, such as management consoles. In short, the references simply do not suggest the desirability, and thus the obviousness, of combining the references. The Final Office Action provides no basis as for why combining the GUI panels that work with Patrizio's ServiceManager software would be desired by Agnihotri, or vice versa. In fact, Appellants respectfully submit that such a combination would simply teach that multiple property sheets and configuration files would be included in Agnihotri's installation wizard and would not teach or suggest the use of console algorithms and generic management objects that could be used to create extension files that would be adapted for use by any of the consoles described by Agnihotri. Appellants' assertion is based on the fact that neither reference teaches "receiving a generic management object..." or "creating an extension file adapted to perform the generic management object on the selected console interface."

Finally, in light of the shortcomings of both of the cited references, Appellants respectfully submit that the Final Office Action used impermissible hindsight in rejecting Appellants' claims. Appellants' contention is based on the fact that the references were selected even though neither reference suggests a desirability of combining the references. Patrizio teaches a system and method for a table row selection in a GUI display while Agnihotri teaches

PATENT

an in-context launch wrapper for integrating applications into management consoles. Evidently, the only reason that these references were selected was because the Examiner used Appellants' claim limitations as "guideposts" in selecting the references. Consequently, impermissible hindsight was used in the rejection of Appellants' claims.

Based on the foregoing, Appellants respectfully submit that the rejection of each of Appellants' independent claims 1, 10, and 19 over Patrizio in view of Agnihotri has been overcome. Therefore, claims 1, 10, and 19 are allowable over Patrizio in view of Agnihotri. Claims 2-9, 11-18, and 20-27 each depends, directly or indirectly, on claims 1, 10, and 19, respectively. Therefore, each of these claims is allowable over Patrizio in view of Agnihotri for at least the same reasons that the independent claims are allowable.

3. Claims 7, 16, 25, and 29 Are Not Obvious And Are Therefore Patentable Over Patrizio In View Of Agnihotri And Cramon

Claims 7, 16, and 25 depend from independent claims 1, 10, and 19, respectively, and are therefore allowable for at least the reasons discussed above. Independent claim 29 includes the limitations found in independent claim 1, and is also patentable for at least the reasons discussed above. Notwithstanding the allowability of claims 7, 16, 25, and 29 as set forth above, claims 7, 16, 25, and 29 include limitations directed at national language. The Final Office Action admits that neither Patrizio nor Agnihotri teaches use of national languages, however the Final Office Action contends that Cramon discloses national languages plug-ins. Once again, the Final Office Action provides no basis, found in the references themselves, for the desirability, and therefore obviousness, of combining Cramon with Patrizio and Agnihotri. Therefore, Appellants respectfully submit that the Final Office Action used the limitations of claims 7, 16, 25, and 29 as guideposts in identifying the Cramon reference.

Further, Cramon teaches a "generic transaction server" for use in e-Business transactions and never mentions or suggests the use of management consoles. Instead, the system described by Cramon operates over a network, such as the Internet. Furthermore, nowhere does Cramon teach or suggest management objects, such as a MOF file, or management definition objects. In light of the fact that there is absolutely no suggestion for combining the e-Business transaction server of Cramon with either the GUI panels that work with Patrizio's ServiceManager software or Agnihotri's installation wizard, Appellants respectfully submit that the combination of Cramon,

PATENT

Patrizio, and Agnihotri is improper as the Final Office Action used impermissible hindsight in forming the combination.

In addition, the cited sections of Cramon do not teach or suggest the elements of Appellants' claims. In paragraph 0214, Cramon notes that "[m]arketing can build and maintain a structured documentation repository (in national language), as a combination of web pages and multimedia files (such as Adobe Acrobat)." Paragraph 0227 discloses a "Frequently Asked Questions" facility, with national language support. Finally, paragraph 0381 discloses a "[c]ountry specific catalogue, with specific national language and prices/currencies." None of the cited sections of Cramon has anything to do with, for example, "converting the user interface data to one or more national languages," or "generating one or more graphical user interfaces using the converted user interface data retrieved from the national language file," as taught and claimed by Appellants. Furthermore, neither Patrizio nor Agnihotri teach or suggest these elements of Appellants' claims.

None of the cited prior art, either separately or in combination, teaches or suggests the elements claimed in claims 7, 16, 25, and 29. For the reasons set forth above, Appellants respectfully submit that claims 7, 16, 25, and 29 are patentable over Patrizio in view of Agnihotri and Cramon.

4. Claims 28, 30, and 31 Are Not Obvious And Are Therefore Patentable Over Patrizio In View Of Agnihotri

Independent claims 28, 30, and 31 include the elements found in independent claims 1, 10, and 19, and are therefore patentable for at least the reasons discussed above with regard to independent claims 1, 10, and 19. In addition, independent claims 28, 30, and 31 include several additional elements, which are neither taught nor suggested by Patrizio in view of Agnihotri. For example, independent claims 28, 30, and 31 each include the element of "determining a layout format based on the number of located second associations." The Final Office Action cites Patrizio at paragraphs 0051-0060 and paragraphs 0068-0070 as disclosing this aspect of Appellants' claims (see Final Office Action, page 6, lines 3-5). Appellants respectfully disagree. The cited sections of Patrizio have to do with how tables render data (see Patrizio, paragraph 0068). The Final Office Action asserts that "all class component [sic] with GUI identifiers read on determining a layout" (see Final Office Action, page 6, line 5). However, Appellants are not

PATENT

simply claiming “determining a layout.” Rather, Appellants teach and claim “determining a layout format based on the number of located second associations.” While Patrizio may disclose rendering data using a table, Patrizio simply does not teach or suggest determining a layout based on the number of located second associations. Further, note that the “second associations” claimed by Appellants are specifically claimed as “second associations corresponding to the identified target class.” All words in the claims must be considered. Therefore, Appellants respectfully submit that using a table to render data is not the same as determining a layout format based on the number of located second associations, where the located second associations correspond to the identified target class, as taught and claimed by Appellants. For the reasons set forth above, Appellants respectfully submit that claims 28, 30, and 31 are patentable over Patrizio in view of Agnihotri.

5. Claim 32 Is Not Obvious And Is Therefore Patentable Over Patrizio In View Of Agnihotri

Independent claim 32 includes the limitations found in independent claim 19, and is therefore patentable for at least the reasons discussed above with regard to independent claim 19. In addition, independent claim 32 includes several additional elements, which are neither taught nor suggested by Patrizio in view of Agnihotri. For example, independent claim 32 includes the following elements:

- means for creating a tree node displayable on a management console corresponding to the selected console interface;
- means for creating a popup menu corresponding to the tree node, wherein the popup menu includes one or more menu selections; and
- means for associating a method to each of the menu selections

The Final Office Action cites Patrizio's Figures 1 and 5-7 as teaching these aspects of Appellants' claims. Even assuming, for the sake of argument, that Patrizio's Figure 1 does show a tree node displayable on a management console (and Appellants do not agree with this assumption), Patrizio certainly does not teach or suggest “a popup menu corresponding to the tree node, wherein the popup menu includes one or more menu selections,” and “associating a method to each of the menu selections,” as taught and claimed by Appellants. Figures 5 through

PATENT


7 of Patrizio disclose that a user can select a particular node in order to populate a lower packages table (see Patrizio, paragraph 0036). The user can further "modify some aspects of the GUI layout while the application is running" (see Patrizio, paragraph 0037).

Appellants are at a loss to understand how selecting a node that populates a packages table, and possibly modifying a GUI layout while an application is running has anything whatsoever to do with "creating a popup menu . . . wherein the popup menu includes one or more menu selections," as taught and claimed by Appellants. Patrizio does not appear to teach or suggest any type of "popup menu" and certainly not a popup menu that "includes or more menu selections." Further, Patrizio does not teach or suggest "associating a method to each of the menu selections." Selecting a node that populates a package table and modifying a GUI layout while an application is running is simply not analogous to "associating a method" to menu selections. Again, Appellants are at a loss to understand where Patrizio associates a method with anything, much less with menu selections that are included as part of a popup menu. For the reasons set forth above, Appellants respectfully submit that claim 32 is patentable over Patrizio in view of Agnihotri.

Conclusion

For the foregoing reasons, Appellants respectfully submit that claims 1-32 are patentable, and, accordingly, Appellant respectfully request that the Examiner's claim rejections be reversed and claims 1-32 be allowed.

Respectfully submitted,

By 
Leslie A. Van Leeuwen, Reg. No. 42,196
Van Leeuwen & Van Leeuwen
Attorneys for Appellants
Telephone: (512) 301-6738
Facsimile: (512) 301-6742

PATENT

I. APPENDIX OF CLAIMS

1. A method of converting management models to one or more console interfaces, said method comprising:
receiving a console selection corresponding to one of the console interfaces;
identifying one or more console algorithms corresponding to the console selection;
retrieving a generic management object from a management definition object; and
creating an extension file adapted to perform the generic management object on the selected console interface.
2. The method as described in claim 1 wherein the management definition object includes a common information model managed object format file.
3. The method as described in claim 1 wherein the extension file includes a plug-in file adapted to interface with the selected console interface.
4. The method as described in claim 1 wherein the retrieving further includes:
identifying one or more object classes and one or more object associations that include the object classes;
selecting one of the object classes;
selecting one of the associations that include the object class, wherein the association includes a simple association;
identifying one or more target classes corresponding to the selected association;
locating one or more second associations corresponding to the identified target class; and
determining a layout format based on the number of located second associations.
5. The method as described in claim 4 further comprising:
displaying the target class name in response to determining that there is one second association; and
displaying the association role in response to determining that there are more than one second associations.

PATENT

6. The method as described in claim 1 further comprising:
retrieving user interface data from the management definition object; and
generating one or more graphical user interfaces using the retrieved user interface data.
7. The method as described in claim 1 further comprising:
retrieving user interface data from the management definition object;
converting the user interface data to one or more national languages;
storing the converted user interface data in one or more national language files;
selecting one of the national languages;
retrieving one of the national language files corresponding to the selected national language; and
generating one or more graphical user interfaces using the converted user interface data retrieved from the national language file.
8. The method as described in claim 1 further comprising:
creating a tree node displayable on a management console corresponding to the selected console interface;
creating a popup menu corresponding to the tree node, wherein the popup menu includes one or more menu selections; and
associating a method to each of the menu selections.
9. The method as described in claim 1 wherein the generic management object includes a display panel that displays information regarding a program product, wherein the management definition object is adapted to manage the program product.
10. An information handling system comprising:
one or more processors;
a memory accessible by the processors;
a nonvolatile storage area accessible by the processors; and
a conversion tool for converting generic management data to specific console interfaces, the conversion tool including:

PATENT

receiving logic for receiving a console selection corresponding to one of the console interfaces;
identification logic for identifying one or more console algorithms corresponding to the console selection;
retrieval logic for receiving a generic management object from a management definition object; and
creation logic for creating an extension file adapted to perform the generic management object on the selected console interface.

11. The information handling system as described in claim 10 wherein the management definition object includes a common information model managed object format file.
12. The information handling system as described in claim 10 wherein the extension file includes a plug-in file adapted to interface with the selected console interface.
13. The information handling system as described in claim 10 wherein the retrieval logic further includes:
identification logic for identifying one or more object classes and one or more object associations that include the object classes;
selection logic for selecting one of the object classes and for selecting one of the associations that include the object class, wherein the association includes a simple association;
identification logic for identifying one or more target classes corresponding to the selected association;
location logic for locating one or more second associations corresponding to the identified target class; and
determination logic for determining a layout format based on the number of located second associations.

PATENT

14. The information handling system as described in claim 13 further comprising:
display logic for displaying the target class name in response to determining that there is one second association and for displaying the association role in response to determining that there are more than one second associations.
15. The information handling system as described in claim 10 further comprising:
retrieval logic for retrieving user interface data from the management definition object;
and
code generation logic for generating one or more graphical user interfaces using the retrieved user interface data.
16. The information handling system as described in claim 10 further comprising:
retrieval logic for retrieving user interface data from the management definition object;
conversion logic for converting the user interface data to one or more national languages;
storage logic for storing the converted user interface data in one or more national language files;
selection logic for selecting one of the national languages;
retrieval logic for retrieving one of the national language files corresponding to the selected national language; and
code generation logic for generating one or more graphical user interfaces using the converted user interface data retrieved from the national language file.
17. The information handling system as described in claim 10 further comprising:
display logic for displaying a tree node on a management console corresponding to the selected console interface; and
display logic for displaying a popup menu corresponding to the tree node, wherein the popup menu includes one or more menu selections, and wherein each menu selection includes an associated method.
18. The information handling system as described in claim 10 wherein the generic management object includes a display panel that displays information regarding a

PATENT

program product, wherein the management definition object is adapted to manage the program product.

19. A computer program product stored on a computer operable medium for converting management models to one or more console interfaces, said computer program product comprising:
 - means for receiving a console selection corresponding to one of the console interfaces;
 - means for identifying one or more console algorithms corresponding to the console selection;
 - means for retrieving a generic management object from a management definition object;
 - and
 - means for creating an extension file adapted to perform the generic management object on the selected console interface.
20. The computer program product as described in claim 19 wherein the management definition object includes a common information model managed object format file.
21. The computer program product as described in claim 19 wherein the extension file includes a plug-in file adapted to interface with the selected console interface.
22. The computer program product as described in claim 19 wherein the retrieving further includes:
 - means for identifying one or more object classes and one or more object associations that include the object classes;
 - means for selecting one of the object classes;
 - means for selecting one of the associations that include the object class, wherein the association includes a simple association;
 - means for identifying one or more target classes corresponding to the selected association;
 - means for locating one or more second associations corresponding to the identified target class; and

PATENT

means for determining a layout format based on the number of located second associations.

23. The computer program product as described in claim 22 further comprising:
means for displaying the target class name in response to determining that there is one second association; and
means for displaying the association role in response to determining that there are more than one second associations.
24. The computer program product as described in claim 19 further comprising:
means for retrieving user interface data from the management definition object; and
means for generating one or more graphical user interfaces using the retrieved user interface data.
25. The computer program product as described in claim 19 further comprising:
means for retrieving user interface data from the management definition object;
means for converting the user interface data to one or more national languages;
means for storing the converted user interface data in one or more national language files;
means for selecting one of the national languages;
means for retrieving one of the national language files corresponding to the selected national language; and
means for generating one or more graphical user interfaces using the converted user interface data retrieved from the national language file.
26. The computer program product as described in claim 19 further comprising:
means for creating a tree node displayable on a management console corresponding to the selected console interface;
means for creating a popup menu corresponding to the tree node, wherein the popup menu includes one or more menu selections; and
means for associating a method to each of the menu selections.

PATENT

27. The computer program product as described in claim 19 wherein the generic management object includes a display panel that displays information regarding a program product, wherein the management definition object is adapted to manage the program product.
28. A method of converting management models to one or more console interfaces, said method comprising:
receiving a console selection corresponding to one of the console interfaces;
identifying one or more console algorithms corresponding to the console selection;
retrieving a generic management object from a management definition object;
creating an extension file adapted to perform the generic management object on the selected console interface;
identifying one or more object classes and one or more object associations that include the object classes;
selecting one of the object classes;
selecting one of the associations that include the object class, wherein the association includes a simple association;
identifying one or more target classes corresponding to the selected association;
locating one or more second associations corresponding to the identified target class; and
determining a layout format based on the number of located second associations.
29. A method of converting management models to one or more console interfaces, said method comprising:
receiving a console selection corresponding to one of the console interfaces;
identifying one or more console algorithms corresponding to the console selection;
retrieving a generic management object from a management definition object;
creating an extension file adapted to perform the generic management object on the selected console interface;
retrieving user interface data from the management definition object;
converting the user interface data to one or more national languages;
storing the converted user interface data in one or more national language files;
selecting one of the national languages;

PATENT

retrieving one of the national language files corresponding to the selected national language; and
generating one or more graphical user interfaces using the converted user interface data retrieved from the national language file.

30. An information handling system comprising:
- one or more processors;
 - a memory accessible by the processors;
 - a nonvolatile storage area accessible by the processors; and
 - a conversion tool for converting generic management data to specific console interfaces, the conversion tool including:
 - receiving logic for receiving a console selection corresponding to one of the console interfaces;
 - identification logic for identifying one or more console algorithms corresponding to the console selection;
 - retrieval logic for receiving a generic management object from a management definition object; and
 - creation logic for creating an extension file adapted to perform the generic management object on the selected console interface;
 - identification logic for identifying one or more object classes and one or more object associations that include the object classes;
 - selection logic for selecting one of the object classes and for selecting one of the associations that include the object class, wherein the association includes a simple association;
 - identification logic for identifying one or more target classes corresponding to the selected association;
 - location logic for locating one or more second associations corresponding to the identified target class;
 - determination logic for determining a layout format based on the number of located second associations; and

PATENT

display logic for displaying the target class name in response to determining that there is one second association and for displaying the association role in response to determining that there are more than one second associations.

31. A computer program product stored on a computer operable medium for converting management models to one or more console interfaces, said computer program product comprising:
- means for receiving a console selection corresponding to one of the console interfaces;
 - means for identifying one or more console algorithms corresponding to the console selection;
 - means for retrieving a generic management object from a management definition object;
 - means for creating an extension file adapted to perform the generic management object on the selected console interface;
 - means for identifying one or more object classes and one or more object associations that include the object classes;
 - means for selecting one of the object classes;
 - means for selecting one of the associations that include the object class, wherein the association includes a simple association;
 - means for identifying one or more target classes corresponding to the selected association;
 - means for locating one or more second associations corresponding to the identified target class;
 - means for determining a layout format based on the number of located second associations;
 - means for displaying the target class name in response to determining that there is one second association; and
 - means for displaying the association role in response to determining that there are more than one second associations.

PATENT

32. A computer program product stored on a computer operable medium for converting management models to one or more console interfaces, said computer program product comprising:
- means for receiving a console selection corresponding to one of the console interfaces;
 - means for identifying one or more console algorithms corresponding to the console selection;
 - means for retrieving a generic management object from a management definition object;
 - means for creating an extension file adapted to perform the generic management object on the selected console interface;
 - means for creating a tree node displayable on a management console corresponding to the selected console interface;
 - means for creating a popup menu corresponding to the tree node, wherein the popup menu includes one or more menu selections; and
 - means for associating a method to each of the menu selections.

PATENT

J. EVIDENCE APPENDIX

Not applicable.

THIS PAGE BLANK (USPTO)

PATENT

K. RELATED PROCEEDINGS APPENDIX

Not applicable.

THIS PAGE BLANK (USPTO)

Docket No. RSW920010048US1

Page 34 of 34
Barker, et. al. - 10/047,792

Atty Ref. No. R108